11-20-00

Please type a plus sign (+) inside this box → +

PTO/SB/05 (11-00)
Approved for use through 10/31/2002 OMB 0651-0032
U S Patent and Trademark Office U S DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

11/17/00 jc694 U.S. PTO

# UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

| | |
|---|---|
| Attorney Docket No. | 004800.P004 |
| First Inventor | Jack B. Dennis |
| Title | Multi-Thread Peripheral Processing Using Dedicated Peripheral Bus |
| Express Mail Label No. | EL466329138US |

## APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents

ADDRESS TO: Assistant Commissioner for Patents
Box Patent Application
Washington, DC 20231

1. ☒ Fee Transmittal Form (e.g., PTO/SB/17)
(Submit an original and a duplicate for fee processing)

2. ☒ Applicant claims small entity status.
See 37 CFR 1.27.

3. ☒ Specification [Total Pages 28]
(preferred arrangement set forth below)
- Descriptive title of the Invention
- Cross References to Related Applications
- Statement Regarding Fed sponsored R & D
- Reference to sequence listing, a table, or a computer program listing appendix
- Background of the Invention
- Brief Summary of the Invention
- Brief Description of the Drawings (if filed)
- Detailed Description
- Claim(s)
- Abstract of the Disclosure

4. ☒ Drawing(s) (35 U.S.C. 113) [Total Sheets 7]

5. Oath or Declaration [Total Pages 3]
a. ☒ Newly executed (original or copy)
b. ☐ Copy from a prior application (37 C.F.R. § 1.63(d))
(for continuation/divisional with Box 18 completed)
i. ☐ DELETION OF INVENTOR(S)
Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b)

6. ☐ Application Data Sheet. See 37 CFR 1.76

7. ☐ CD-ROM or CD-R in duplicate, large table or Computer Program (Appendix)

8. Nucleotide and/or Amino Acid Sequence Submission
(if applicable, all necessary)
a. ☐ Computer Readable Form (CRF)
b. Specification Sequence Listing on:
i. ☐ CD-ROM or CD-R (2 copies); or
ii. ☐ paper
c. ☐ Statements verifying identity of above copies

jc658 U.S. PTO 09/715772 11/17/00

## ACCOMPANYING APPLICATION PARTS

9. ☒ Assignment Papers (cover sheet & document(s))
10. ☐ 37 C.F.R. § 3.73(b) Statement (when there is an assignee) ☐ Power of Attorney
11. ☐ English Translation Document (if applicable)
12. ☐ Information Disclosure Statement (IDS)/PTO-1449 ☐ Copies of IDS Citations
13. ☐ Preliminary Amendment
14. ☐ Return Receipt Postcard (MPEP 503)
(Should be specifically itemized)
15. ☐ Certified Copy of Priority Document(s)
(if foreign priority is claimed)
16. ☐ Request and Certification under 35 U.S.C. 122 (b)(2)(B)(i). Applicant must attach form PTO/SB/35 or its equivalent.
17. ☒ Other: Request to Add Additional Inventors

18. If a CONTINUING APPLICATION, check appropriate box, and supply the requisite information below and in a preliminary amendment:
☐ Continuation ☐ Divisional ☐ Continuation-in-part (CIP) of prior application No: _____

Prior application Information: Examiner _____ Group/Art Unit: _____

**For CONTINUATION OR DIVISIONAL APPS only: The entire disclosure of the prior application, from which an oath or declaration is supplied under Box 5b, is considered a part of the disclosure of the accompanying continuation or divisional application and is hereby incorporated by reference. The incorporation can only be relied upon when a portion has been inadvertently omitted from the submitted application parts.**

## 18. CORRESPONDENCE ADDRESS

☒ Customer Number or Bar Code Label 08791
PATENT TRADEMARK OFFICE
(Insert Customer No. or Attach bar code label here)

or ☐ Correspondence address below

| | | | | | |
|---|---|---|---|---|---|
| Name | | | | | |
| Address | | | | | |
| City | | State | | Zip Code | |
| Country | | Telephone | | Fax | |

| | | | |
|---|---|---|---|
| Name (Print/Type) | Thinh V. Nguyen | Registration No. (Attorney/Agent) | 42,034 |
| Signature | | Date | 11/17/00 |

Burden Hour Statement This form is estimated to take 0.2 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, U S Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS SEND TO Assistant Commissioner for Patents, Box Patent Application, Washington, DC 20231.

PTO/SB/17 (08-00)
Approved for use through 10/31/2002 OMB 0651-0032
U S Patent and Trademark Office U S DEPARTMENT OF COMMERCE


# FEE TRANSMITTAL for FY 2000

*Patent fees are subject to annual revision*

**TOTAL AMOUNT OF PAYMENT** ($) 723.00

| *Complete if Known* | |
|---|---|
| Application Number | |
| Filing Date | November 17, 2000 |
| First Named Inventor | Jack B. Dennis |
| Examiner Name | |
| Group/Art Unit | |
| Attorney Docket No. | 004800.P004 |

## METHOD OF PAYMENT (check one)

1. ☒ The Commissioner is hereby authorized to charge indicated fees and credit any overpayments to:

Deposit Account Number: 02-2666

Deposit Account Name: Blakely, Sokoloff, Taylor & Zafman LLP

☒ Charge Any Additional Fee(s) Required Under 37 CFR §§ 1 16, 1 17, 1 18 and 1 20.

☒ Applicant claims small entity status See 37 CFR 1.27

2. ☒ **Payment Enclosed:**

☒ Check ☐ Credit card ☐ Money Order ☐ Other

## FEE CALCULATION

### 1. BASIC FILING FEE

| Large Entity Fee Code | Large Entity Fee ($) | Small Entity Fee Code | Small Entity Fee ($) | Fee Description | Fee Paid |
|---|---|---|---|---|---|
| 101 | 710 | 201 | 355 | Utility filing fee | 355.00 |
| 106 | 320 | 206 | 160 | Design filing fee | |
| 107 | 490 | 207 | 245 | Plant filing fee | |
| 108 | 710 | 208 | 355 | Reissue filing fee | |
| 114 | 150 | 214 | 75 | Provisional filing fee | |

**SUBTOTAL (1)** ($) 355.00

### 2. EXTRA CLAIM FEES

| | | | Extra Claims | | Fee from below | | |
|---|---|---|---|---|---|---|---|
| Total Claims | 42 | - 20** = | 22 | X | 9.00 | = | $198.00 |
| Independent Claims | 5 | - 3** = | 2 | X | 40.00 | = | $80.00 |
| Multiple Dependent | | | | | | = | |

**or number previously paid, if greater, For Reissues, see below

| Large Entity Fee Code | Large Entity Fee ($) | Small Entity Fee Code | Small Entity Fee ($) | Fee Description |
|---|---|---|---|---|
| 103 | 18 | 203 | 9 | Claims in excess of 20 |
| 102 | 80 | 202 | 40 | Independent claims in excess of 3 |
| 104 | 130 | 204 | 135 | Multiple Dependent claim, if not paid |
| 109 | 80 | 209 | 40 | **Reissue independent claims over original patent |
| 110 | 18 | 210 | 9 | **Reissue claims in excess of 20 and over original patent |

**SUBTOTAL (2)** ($) 278.00

## FEE CALCULATION (continued)

### 3. ADDITIONAL FEE

| Large Entity Fee Code | Large Entity Fee ($) | Small Entity Fee Code | Small Entity Fee ($) | Fee Description | Fee Paid |
|---|---|---|---|---|---|
| 105 | 130 | 205 | 65 | Surcharge - late filing fee or oath | |
| 127 | 50 | 227 | 25 | Surcharge - late provisional filing fee or cover sheet. | |
| 139 | 130 | 139 | 130 | Non-English specification | |
| 147 | 2,520 | 147 | 2,520 | For filing a request for reexamination | |
| 112 | 920* | 112 | 920* | *Requesting publication of SIR prior to Examiner action | |
| 113 | 1,840* | 113 | 1,840* | *Requesting publication of SIR after Examiner action | |
| 115 | 110 | 215 | 55 | Extension for response within first month | |
| 116 | 390 | 216 | 195 | Extension for response within second month | |
| 117 | 890 | 217 | 445 | Extension for response within third month | |
| 118 | 1,390 | 218 | 695 | Extension for response within fourth month | |
| 128 | 1,890 | 228 | 945 | Extension for response within fifth month | |
| 119 | 310 | 219 | 155 | Notice of Appeal | |
| 120 | 310 | 220 | 155 | Filing a brief in support of an appeal | |
| 121 | 270 | 221 | 135 | Request for oral hearing | |
| 138 | 1,510 | 138 | 1,510 | Petition to institute a public use proceeding | |
| 140 | 110 | 240 | 55 | Petition to revive - unavoidable | |
| 141 | 1,240 | 241 | 620 | Petition to revive - unintentional | |
| 142 | 1,240 | 242 | 620 | Utility issue fee (or reissue) | |
| 143 | 440 | 243 | 220 | Design issue fee | |
| 144 | 600 | 244 | 300 | Plant issue fee | |
| 122 | 130 | 122 | 130 | Petitions to the Commissioner | |
| 123 | 130 | 123 | 130 | Petitions related to provisional applications | |
| 126 | 180 | 126 | 180 | Submission of Information Disclosure Stmt | |
| 581 | 40 | 581 | 40 | Recording each patent assignment per property (times number of properties) | 40.00 |
| 146 | 710 | 246 | 355 | Filing a submission after final rejection (37 CFR § 1.129(a)) | |
| 149 | 710 | 249 | 355 | For each additional invention to be examined (37 CFR § 1.129(b)) | |
| 179 | 710 | 279 | 355 | Request for Continued Examination (RCE) | |
| 169 | 900 | 169 | 900 | Request for expedited examination of a design application | |
| Other fee (specify) | | | | Request to Add Additional Inventor | 50.00 |
| Other fee (specify) | | | | | |

*Reduced by Basic Filing Fee Paid

**SUBTOTAL (3)** ($) 90.00

## SUBMITTED BY

Complete (if applicable)

| Name (Print/Type) | Thinh V. Nguyen | Registration No. (Attorney/Agent) | 42,034 | Telephone | (714) 557-3800 |
|---|---|---|---|---|---|
| Signature | | | | Date | 11/17/00 |

**WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2039.**

Our Ref. No. 004800.P004
Express Mail No.: EL466329138US

UNITED STATES PATENT APPLICATION

FOR

# MULTI-THREAD PERIPHERAL PROCESSING

# USING DEDICATED PERIPHERAL BUS

INVENTORS:

Jack B. Dennis
Sam B. Sandbote

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 Wilshire Blvd., 7th Floor
Los Angeles, CA 90025-1026
(714) 557-3800

# MULTI-THREAD PERIPHERAL PROCESSING USING DEDICATED PERIPHERAL BUS

## RELATED APPLICATION

This application claims the benefit of U.S. Provisional Application No. 60/166,686, titled "Integrated Processor for Multithread with Real-Time Input-Output Capability" filed on November 19, 1999.

## BACKGROUND

### 1. Field of the Invention

This invention relates to computer architecture. In particular, the invention relates to multi-thread computers.

### 2. Description of Related Art

Demand in high speed data transmission has given rise to many large bandwidth network protocols and standards. For example, the Synchronous Optical Network (SONET) has a number of standards used in Wide Area Network (WAN) with speeds ranging from a few megabits per second (Mbps) to several gigabits per second (Gbps). Popular standards include T1 (1.5 Mbps), T3 (45 Mbps), OC-3c (155 Mbps), OC-12c (622 Mbps), OC-48c (2.5 Gbps), OC-192c (10 Gbps), OC-768c (40 Gbps), etc.

In network applications, the requirements for cell processing and packet processing functions at line rates for broadband communications switches and routers have become increasingly difficult to satisfy, and demand multiple processor configurations to meet performance requirements. The processing of cells and packets typically requires frequent interactions with special function

devices and external units. Existing techniques are inadequate to meet real-time demands without degrading performance.

Therefore, there is a need to have a technique to perform peripheral operations for cell and packet processing.

## BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is a diagram illustrating a system in which one embodiment of the invention can be practiced.

Figure 2 is a diagram illustrating a multiprocessor core shown in Figure 1 according to one embodiment of the invention.

Figure 3 is a diagram illustrating a multi-threaded processor shown in Figure 2 according to one embodiment of the invention.

Figure 4 is a diagram illustrating a processing slice shown in Figure 3 according to one embodiment of the invention.

Figure 5A is a diagram illustrating format of a command message according to one embodiment of the invention.

Figure 5B is a diagram illustrating format of a response message according to one embodiment of the invention.

Figure 6 is a diagram illustrating peripheral operations according to one embodiment of the invention.

Figure 7 is a diagram illustrating an instruction processing according to one embodiment of the invention.

## DESCRIPTION

The present invention is a method and apparatus to perform peripheral operations in a multi-thread processor. A peripheral bus is coupled to a peripheral unit to transfer peripheral information including a command message specifying a peripheral operation. A processing slice executes a plurality of threads and is coupled to the peripheral bus to execute peripheral operations. The plurality of threads includes a first thread sending the command message to the peripheral unit.

The command message includes at least one of a message content, a peripheral address identifying the peripheral unit, and a command code specifying the peripheral operation. The peripheral information includes a response message sent from the peripheral unit to the processing slice. The response message indicates the peripheral operation is completed and includes at least one of a thread identifier identifying the first thread, one or more result phrases, each including an operation result, a data register address specifying a data register in the processing slice to store the operation result, and an end flag indicating the last one of the result phrases.

The command message may be a wait instruction or a non-wait instruction. The processing slice disables the first thread after sending the command message if the command message is a wait instruction. The first thread continues to execute after sending the command message if the command message is a non-wait instruction. The processing slice enables the first thread after receiving the response message from the peripheral unit if the first thread was disabled.

The peripheral bus may include a bi-directional bus to transfer the command message from the processing slice to the peripheral unit and the response message from peripheral unit to the processing slice.

The processing slice includes an instruction processing unit, a thread control unit, a peripheral unit, a memory access unit, a functional unit, a condition

code memory, and a register file. The processing slice is configured to support execution of a number of threads. Several processing slices operate concurrently in a processor. A multiprocessor core may include several of these processors in a network processor system.

In the processing slice, the instruction processing unit processes instructions fetched from a program memory. The instruction processing unit includes an instruction fetch unit, an instruction buffer, and an instruction decoder and dispatcher. The instruction fetch unit fetches the instructions from the program memory using a number of program counters, each of which corresponds to each of the threads. The instruction buffer holds the fetched instructions waiting for execution. The instruction decoder and dispatcher decodes the instructions and dispatches the decoded instructions to the memory access unit, the functional unit, or the peripheral unit.

The thread control unit manages initiation and termination of at least one of the threads. The peripheral unit transfers the peripheral information between the peripheral unit and the instruction processing unit. The peripheral unit receives command messages from the processing slices to direct performance of peripheral operations and sends response messages to the processing slices to convey results of the peripheral operations to the threads. The memory access unit provides access to one of a number of data memories via a data memory switch. The functional unit performs an operation specified in one of the instructions. The condition code memory stores a number of condition codes, each of which corresponds to each of the threads. The register file has a number of data registers, of which a subset is associated with each of the threads.

In the following description, for purposes of explanation, numerous details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that these specific details are not required in order to practice the present invention. In other instances, well-

known electrical structures and circuits are shown in block diagram form in order not to obscure the present invention.

Figure 1 is a diagram illustrating a system 100 in which one embodiment of the invention can be practiced. The system 100 includes a multiprocessor core 110, a memory controller 120, peripheral units 130, an off-chip program/data memory 140, and a host control processor 150.

The multiprocessor core 110 is a high-performance multi-thread computing subsystem capable of performing all functions related to network operations. These network operations may include adjusting transmission rates, handling special cells and packets used to implement flow control protocols on an individual connection basis, and supporting Asynchronous Transfer Mode (ATM) traffic management for Available Bit Rate (ABR), Variable Bit Rate (VBR), and Unspecified Bit Rate (UBR) connections. The memory controller 120 provides access to additional memory devices and includes circuitry to interface to various memory types including dynamic random access memory (DRAM) and static random access memory (SRAM). The peripheral units 130 include a number of peripheral or input/output (I/O) units for peripheral or I/O operations. The peripheral units 130 include an input interface 162, and output interface 164, a cyclic redundancy code (CRC) engine 166, a check-out content addressable memory (CAM) 168, a bit vector unit 172, and a spare 174. The input and output interfaces 162 and 164 provide interfaces to inbound and outbound network traffics, respectively. These interfaces may include line and switch/system interfaces that support industry standards, including multi-phy features such as Universal Test and Operations PHY Interface for ATM (UTOPIA). The CRC engine 166 supports segmentation and re-assembly for ATM Adaptation Layer Type 5 (AAL5) transmission of packets over ATM connections. The check-out CAM 168 is an associative memory unit that supports the maintenance of several connection records in the on-chip memory for the duration of cell processing for those connections. The bit vector unit 172 supports round-robin scheduling algorithms at the OC-48 line rate.

The off-chip program/data memory 140 includes memory devices that store programs or data in addition to the on-chip programs and data stored in the multiprocessor core 110. The host control processor 150 is a processor that performs the general control functions in the network. These functions may include connection set-up, parameter adjustment, operation monitoring, program loading and debugging support.

Figure 2 is a diagram illustrating the multiprocessor core 110 shown in Figure 1 according to one embodiment of the invention. The multiprocessor core 110 includes four multi-thread processors $210_1$ to $210_4$, a split transaction switch 220, a host interface bus 250, and a peripheral bus 260. It is noted that the use of four processors is for illustrative purposes only. As is known to one skilled in the art, any reasonable number of processors can be used.

The four multi-thread processors $210_1$ to $210_4$ are essentially the same. Each of the processors $210_1$ to $210_4$ has local program and data memories for N-bit words of instructions and data, respectively. In one embodiment, N=32. The split transaction switch 210 permits each of the processors to access the data words held in any of the other three data memories with a small additional access time.

The host interface bus 250 allows the any of the four processors $210_1$ to $210_4$ to communicate with the host control processor 150 (Figure 1). This includes passing parameters, loading program and data, and reporting status. The peripheral bus 260 allows any one of the peripheral units 130 to communicate with any of the processors $210_1$ to $210_4$. Some peripheral units may have direct memory access (DMA) channels to the local data memories of any one of the processors $210_1$ to $210_4$. In one embodiment, each of these channels supports burst transfer of 32-bit data at 100 MHz clock rate, equivalent to greater than the OC-48 speed.

Figure 3 is a diagram illustrating the multi-thread processor 210 shown in Figure 2 according to one embodiment of the invention. The multi-thread

processor 210 includes four processing slices (PS's) $310_1$ to $310_4$, a data memory switch 320, banks of data memory 330, a peripheral message unit 340, a control and monitor interface 350, and a program memory 360. It is noted that the use of four PS's is for illustrative purposes only. As is known by one skilled in the art, any number of PS's can be used.

The multi-thread processor 210 is a data and/or information processing machine that supports the simultaneous execution of several programs, each program being represented by a sequence of instructions. A thread is a sequence of instructions that may be a program, or a part of a program. The multi-thread processor 210 may have one or more instruction execution resources such as arithmetic logic units, branch units, memory interface units, and input-output interface units. In any operation cycle of the multi-thread processor 210, any instruction execution resource may operate to carry out execution of an instruction in any thread. Any one instruction resource unit may participate in the execution of instructions of different threads in successive cycles of processor operation. To support this mode of operation, the multi-thread processor 210 may have a separate hardware register, referred to as the program counter, for each thread that indicates the position or address of the next instruction to be executed within the thread. A multi-thread multiprocessor is a data and/or information processing system composed of several multi-thread processors.

Each of the PS's $310_1$ to $310_4$ contains a program sequencer and execution units to perform instruction fetch, decode, dispatch and execution for four threads. Each of the PS's operates by interleaving the execution of instructions from the four threads, including the ability to execute several instructions concurrently in the same clock cycle. The data memory switch 320 allows any of the four PS's $310_1$ to $310_4$ to access any data memory bank in the banks of data memories 330. The banks of memories 330 include four banks $335_1$ to $335_4$: data memory banks 0 to 3. Each of the data memory banks $335_1$ to $335_4$ stores data to be used or accessed by any of the PS's $310_1$ to $310_4$. In addition, each of the data memory banks $335_1$ to $335_4$ has an interface to the DMA bus to support DMA transfers

between the peripherals and data memory banks. The banks $335_1$ to $335_4$ are interleaved on the low-order address bits. In this way, DMA transfers to and from several of the peripheral units 130 can proceed simultaneously with thread execution without interference.

The four PS's $310_1$ to $310_4$ are connected to the peripheral message unit 340 via four PS buses $315_1$ to $315_4$, respectively. The peripheral message unit 340 is a distribution or switching location to switch the peripheral bus 260 to each of the PS buses $315_1$ to $315_4$. The peripheral message unit 340 is interfaced to the peripheral bus 260 via a command bus 342 and a response bus 344. The command bus 342 and the response bus 344 may be combined into one single bi-directional bus. Appropriate signaling scheme or handshaking protocol is used to determine if the information is a command message or the response message.

When a thread in any of the four PS's $310_1$ to $310_4$ executes a wait or no_wait instruction for a peripheral operation, a command message is sent from the issuing PS to the command bus 342. The command message specifies the peripheral unit where the peripheral operation is to be performed by including the address of the peripheral unit. All peripheral units connected to the peripheral bus 260 have an address decoder to decode the peripheral unit address in the command message. When a peripheral unit recognizes that it is the intended peripheral unit for the peripheral operation, it will decode the command code contained in the command message and then carry out the operation. If the command message is a wait message instruction, the issuing thread is stalled for an interval during which the responding peripheral unit carries out the peripheral operation. During this interval, the resources associated with the issuing thread are available to other threads in the issuing slice. In this way, high resource utilization can be achieved. If it is a no_wait instruction, the issuing thread continues executing its sequence without waiting for the peripheral operation to be completed. The issuing thread may or may not need a response from the peripheral unit.

The control and monitor interface 350 permits the host control processor 150 to interact with any one of the four PS's $310_1$ to $310_4$ through the host interface bus 350 to perform control and monitoring functions. The program memory 360 stores program instructions to be used by any one of the threads in any one of the four PS's $310_1$ to $310_4$. The program memory 360 supports simultaneous fetches of four instruction words in each clock cycle.

Figure 4 is a diagram illustrating the processing slice 310 shown in Figure 3 according to one embodiment of the invention. The processing slice 310 includes an instruction processing unit 410, a peripheral unit interface 420, a register file 430, a condition code memory 440, a functional unit 450, a memory access unit 460, and a thread control unit 470. The processing slice 310 is configured to have four threads. The use of four threads is for illustrative purposes only. As is known by one skilled in the art, any number of threads can be used.

The instruction processing unit 410 processes instructions fetched from the program memory 360. The instruction processing unit 410 includes an instruction fetch unit 412, an instruction buffer 414, and an instruction decoder and dispatcher 416. The instruction fetch unit 412 fetches the instructions from the program memory 360 using a plurality of program counters. Each program counter corresponds to each of the threads. The instruction buffer 414 holds the fetched instructions waiting for execution for any of the four threads. The instruction decoder and dispatcher 416 decodes the instructions and dispatches the decoded instructions to the peripheral unit 420, the register file 430, the condition code memory 440, the functional unit 450, or the memory access unit 460 as appropriate.

The thread control unit 470 manages initiation and termination of at least one of the four threads. The thread control unit 470 includes program counters 472 and a program (or code) base register unit 473 containing program base addresses corresponding to the threads. Execution of a computation may start

from a single thread, executing the main function of the program. A thread may initiate execution of another thread by means of a start instruction. The new thread executes in the same function context as the given thread. In other words, it uses the same data and code base register contents. A thread runs until it encounters a peripheral wait, or until it reaches a quit instruction.

The peripheral unit interface 420 is connected to the instruction processing unit 410 and the peripheral message unit 340 to transfer the peripheral information between the peripheral units 130 (Figure 1) and the instruction processing unit 410. The peripheral operation may be an input or an output operation. In one embodiment, an input or output operation is initiated by a message instruction that causes a command message to be transferred to a specified peripheral unit over the peripheral bus. The message instruction may be marked wait or no_wait. If the message instruction is marked wait, it is expected that the peripheral unit will return a response message; the processing slice that issued the message-wait instruction will execute the following instructions of that thread only when the response message has been received over the peripheral bus.

In a peripheral operation, a command message includes a content part that contains data words from data registers specified in the message instruction. If a response message is returned, it contains one or more result phrases, each specifying a data word and a data register identifier; the slice puts each data word in the specified data register, and continues execution of the thread after processing the last result phrase.

The register file 430 has four sets of data registers. Each of the four sets of data registers corresponds to each of the four threads. The data registers store data or temporary items used by the threads. Peripheral operations may reference the data registers in the command or response message.

The condition code memory 440 stores four condition codes. Each of the condition codes corresponds to each of the four threads. The condition code includes condition bits that represent the conditions generated by the functional

unit 450. These condition bits include overflow, greater_than, equal, less_than conditions. The condition bits are set according to the type of the instruction being executed. For example, the compare instructions sets the greater_than, equal, and less_than condition bits and clears the overflow condition bit.

The functional unit 450 performs an operation specified in the dispatched instruction. The functional unit 450 performs all operations of the instruction set that manipulate values in the data registers. These operations include arithmetic and logical register operations, shift and selected bit operations. The operation performed by the functional unit 450 is determined by a decoded opcode value passed from the instruction decoder and dispatcher 416. The functional unit 450 has connections to the condition code memory 440 to set a thread's condition code according to the outcome of an arithmetic operation or comparison.

The memory access unit 460 provides for read and write accesses to any of the four data memory banks $315_1$ to $315_4$ via the data memory switch 320 (Figure 3). The memory access unit 460 has a base register unit 462 having four base registers to receive the base address used in address formation and for saving and restoring the base registers for the call and return instructions. Each of the four data base registers corresponds to each of the four threads.

In one alternative embodiment of the invention, the instruction processing unit 410 may include M program base registers. Each of the M program base registers is associated with each of the M threads. The contents of a base register are added to the contents of the corresponding program counter to determine the location in the program memory from which the next instruction for the corresponding thread is to be fetched. An advantage of this scheme is that the branch target specified in the instruction that transfers control may be represented in fewer bits for local transfers.

In one alternative embodiment of the invention, the memory access unit 460 may include a data base register unit 462 having M data base registers 462. Each of the M data base registers is associated with each of the M threads. The

contents of the appropriate base register are added to the corresponding program counter to form the effective address for selected instructions. This permits offset addressing to be used, leading to more compact programs.

Figure 5A is a diagram illustrating format of a command message 500 according to one embodiment of the invention. The command message 500 includes a processor identifier 505, a thread identifier 507, a message content 510, a peripheral address 520, and a command code 530.

The processor identifier 505 and the thread identifier 507 identifies the thread in the processor that issues the command so that the peripheral unit can direct a resulting response message to the correct processor and thread. This is necessary to allow peripheral units to be shared by threads executing on distinct processors.

The message content 510 is a sequence of N-bit words taken from the data registers specified in the message instruction. The message content 510 includes the data or operand to be used by the peripheral unit. For a DMA operation, the message content 510 may contain the starting address for the transfer in data memory, and a count indicating the number of words to be transferred. The peripheral address 520 specifies the address of the peripheral unit to perform the peripheral operation specified by the command code 530. The command code 530 specifies the peripheral operation performed by the peripheral unit corresponding to the address specified in the peripheral address 520.

Figure 5B is a diagram illustrating format of a response message 550 according to one embodiment of the invention. The response message 550 includes a thread identifier 560, K data register address $570_1$ to $570_K$, K operation result $580_1$ to $580_K$, and an end flag 590.

The thread identifier 560 specifies the thread that the response message is directed to. Usually, this thread is the thread that issued the command message to the peripheral unit. Each of the data register addresses $570_1$ to $570_K$ specifies the

data register in the register file 430 to store the corresponding one of K operation results $580_1$ to $580_K$. Each of the operation results $580_1$ to $580_K$ is the result of the corresponding operation, which may include status information regarding the peripheral operation. The end flag 590 indicates the end of the response message 550.

Figure 6 is a diagram 600 illustrating peripheral operations according to one embodiment of the invention. The diagram 600 shows the interactions between the processing slice 310 and the peripheral unit 130. Although only one peripheral unit 130 is shown, it is contemplated that more than one peripheral units may operate with the processing slice 310.

Within the processing slice 310, the begin cycle 610 starts an instruction execution cycle. Block 620 selects one or more instructions for execution. One or more instructions may be selected for execution concurrently. Suppose K instructions are selected for execution. Block $630_1$ to Block $630_K$ process instruction 1 to instruction K, respectively. The details of the block $630_k$, where k = 1, . . , K, are shown in Figure 7. Blocks $630_1$ to $630_K$ may terminate at the same time or at different times at the end cycle 640. Blocks $630_1$ to $630_K$ will interact with the peripheral unit 130 via the peripheral bus 260 via the peripheral bus 260 when they are processing message instructions.

Within the peripheral unit 130, there are processing blocks to receive and send peripheral messages via the peripheral bus 260. Block 650 receives the command message from the processing slice 310. Block 660 starts the I/O operation for the thread T as specified in the command message. Block 670 completes the I/O operation for thread T' where thread T' may or may not be the same as thread T. The command message for thread T' may have been received by block 650 either earlier or later than the command message for thread T, due to out-of-order performance of operations by the peripheral unit. Block 680 sends a response message to indicate the completion of thread T' to the processing slice via the peripheral bus 260.

Figure 7 is a diagram illustrating the instruction processing block $630_k$ shown in Figure 6 according to one embodiment of the invention. In this block, begin and end nodes 710 and 790, respectively, indicate the beginning and the end of the instruction processing block $630_k$.

Begin node 710 starts the instruction processing. Block 720 dispatches thread T according to the instruction type. If the instruction type is a non-message instruction, block 730 processes the instruction as appropriate. If the instruction is a message instruction, block 740 sends the command message to the peripheral unit 130. The command message is to be received by block 650 as shown in Figure 6. If the command message is a wait instruction, block 750 disables thread T and goes to the end node 790. If the command message is a non-wait instruction, block 780 indicates that the instruction is done and goes to the end node 790.

Block 760 receives the response message for thread T' as sent from block 680 shown in Figure 6. Then, block 770 enables thread T' if thread T' was disabled and then goes to block 780 to indicate that the instruction processing is done. Next, end node 790 indicates the end of the instruction processing block.

While one or more threads are waiting for completion of peripheral operations, other threads may continue executing instructions that utilize functional unit and memory resources. In this way, higher average performance of the multi-thread processors may be achieved.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

## CLAIMS

What is claimed is:

1. An apparatus comprising:

a peripheral bus coupled to a peripheral unit to transfer peripheral information including a command message specifying a peripheral operation; and

a processing slice coupled to the peripheral bus to execute a plurality of threads, the plurality of threads including a first thread sending the command message to the peripheral unit.

2. The apparatus of claim 1 wherein the peripheral unit is one of an input device and an output device.

3. The apparatus of claim 1 wherein the peripheral operation is one of an input operation and an output operation.

4. The apparatus of claim 1 wherein the command messages includes at least one of a message content, a peripheral address identifying the peripheral unit, and a command code specifying the peripheral operation.

5. The apparatus of claim 1 wherein the peripheral information includes a response message sent from the peripheral unit to the processing slice, the response message indicating the peripheral operation is completed.

6. The apparatus of claim 5 wherein the response message includes at least one of a thread identifier identifying the first thread, an operation result of the peripheral operation, a data register address specifying a data register in the processing slice to store the operation result, and a length indicator indicating length of the response message.

7. The apparatus of claim 6 wherein the peripheral bus comprises:

a bi-directional bus to transfer the command message from the processing slice to the peripheral unit and the response message from peripheral unit to the processing slice.

8. The apparatus of claim 1 wherein the processing slice disables the first thread after sending the command message if the command message is a wait instruction.

9. The apparatus of claim 1 wherein the first thread continues to execute after sending the command message if the command message is a non-wait instruction.

10. The apparatus of claim 8 wherein the processing slice enables the first thread after receiving the response message from the peripheral unit if the first thread was disabled.

11. The apparatus of claim 1 wherein the processing slice comprises:

an instruction processing unit to process instructions fetched from a program memory; and

a thread control unit coupled to the instruction processing unit to manage initiating and termination of at least one of the plurality of threads.

12. The apparatus of claim 11 wherein the processing slice further comprises:

a memory access unit coupled to the instruction processing unit to provide access to one of a plurality of data memories via a data memory switch, the memory access unit having a plurality of data base registers, each of the data base registers corresponding to each of the threads; and

a functional unit coupled to the instruction processing unit to perform an operation specified in one of the instructions; and

a register file coupled to the instruction processing unit and the peripheral unit having a plurality of data registers, each of the data registers corresponding to each of the threads.

13. The apparatus of claim 12 wherein the instruction processing unit comprises:

an instruction fetch unit to fetch the instructions from the program memory using a plurality of program counters, each program counter corresponding to each of the threads;

an instruction buffer coupled to the instruction fetch unit to hold the fetched instructions; and

an instruction decoder and dispatcher coupled to the instruction buffer to decode the instructions and dispatch the decoded instructions to one of the memory access unit, the functional unit, and the peripheral unit;

wherein the instructions are executed concurrently in a clock cycle.

14. A method comprising:

transferring peripheral information to a peripheral unit via a peripheral bus, the peripheral information including a command message specifying a peripheral operation; and

executing a plurality of threads by a processing slice, the plurality of threads including a first thread sending the command message to the peripheral unit.

15. The method of claim 14 wherein the peripheral unit is one of an input device and an output device.

16. The method of claim 14 wherein the peripheral operation is one of an input operation and an output operation.

17. The method of claim 14 wherein the command messages includes at least one of a message content, a peripheral address identifying the peripheral unit, and a command code specifying the peripheral operation.

18. The method of claim 14 wherein the peripheral information includes a response message sent from the peripheral unit to the processing slice, the response message indicating the peripheral operation is completed.

19. The method of claim 18 wherein the response message includes at least one of a thread identifier identifying the first thread, an operation result of the peripheral operation, a data register address specifying a data register in the processing slice to store the operation result, and a length indicator indicating length of the response message.

20. The method of claim 19 wherein transferring the peripheral information comprises:

transferring the command message from the processing slice to the peripheral unit and the response message from peripheral unit to the processing slice via a bi-directional bus.

21. The method of claim 14 wherein executing the plurality of threads comprises disabling the first thread after sending the command message if the command message is a wait instruction.

22. The method of claim 14 wherein executing the plurality of threads comprises continuing executing the first thread after sending the command message if the command message is a non-wait instruction.

23. The method of claim 21 wherein executing the plurality of threads comprises enabling the first thread after receiving the response message from the peripheral unit if the first thread was disabled.

24. The method of claim 14 wherein executing the plurality of threads comprises:

processing instructions fetched from a program memory by an instruction processing unit;

managing initiating and termination of at least one of the plurality of threads b a thread control unit.

25. The method of claim 24 wherein executing the plurality of threads further comprises:

accessing to one of a plurality of data memories by a memory access unit via a data memory switch, the memory access unit having a plurality of data base registers, each of the data base registers corresponding to each of the threads;

performing an operation specified in one of the instructions by a functional unit; and

storing data in a register file having a plurality of data registers, each of the data registers corresponding to each of the threads.

26. The method of claim 24 wherein processing instructions comprises:

fetching the instructions from the program memory using a plurality of program counters by an instruction fetch unit, each program counter corresponding to each of the threads;

holding the fetched instructions in an instruction buffer;

decoding the instructions and dispatching the decoded instructions by an instruction decoder and dispatcher to one of the memory access unit, the functional unit, and the peripheral unit; and

executing the instructions concurrently in a clock cycle.

27. A processing system comprising:

a plurality of banks of data memory;

a data memory switch coupled to the banks to data memory;

a program memory to store a program;

a peripheral bus coupled to a peripheral unit to transfer peripheral information including a command message specifying a peripheral operation; and

a processing slice coupled to the peripheral bus to execute a plurality of threads, the plurality of threads including a first thread sending the command message to the peripheral unit.

28. The processing system of claim 27 wherein the peripheral unit is one of an input device and an output device.

29. The processing system of claim 27 wherein the peripheral operation is one of an input operation and an output operation.

30. The processing system of claim 27 wherein the command messages includes at least one of a message content, a peripheral address identifying the peripheral unit, and a command code specifying the peripheral operation.

31. The processing system of claim 27 wherein the peripheral information includes a response message sent from the peripheral unit to the processing slice, the response message indicating the peripheral operation is completed.

32. The processing system of claim 31 wherein the response message includes at least one of a thread identifier identifying the first thread, an operation result of the peripheral operation, a data register address specifying a data register in the processing slice to store the operation result, and a length indicator indicating length of the response message.

33. The processing system of claim 32 wherein the peripheral bus comprises:

a bi-directional bus to transfer the command message from the processing slice to the peripheral unit and the response message from peripheral unit to the processing slice.

34. The processing system of claim 27 wherein the processing slice disables the first thread after sending the command message if the command message is a wait instruction.

35. The processing system of claim 27 wherein the first thread continues to execute after sending the command message if the command message is a non-wait instruction.

36. The processing system of claim 34 wherein the processing slice enables the first thread after receiving the response message from the peripheral unit if the first thread was disabled.

37. The processing system of claim 27 wherein the processing slice comprises:

an instruction processing unit to process instructions fetched from a program memory; and

a thread control unit coupled to the instruction processing unit to manage initiating and termination of at least one of the plurality of threads.

38. The processing system of claim 37 wherein the processing slice further comprises:

a memory access unit coupled to the instruction processing unit to provide access to one of a plurality of data memories via a data memory switch, the memory access unit having a plurality of data base registers, each of the data base registers corresponding to each of the threads;

a functional unit coupled to the instruction processing unit to perform an operation specified in one of the instructions; and

a register file coupled to the instruction processing unit and the peripheral unit having a plurality of data registers, each of the data registers corresponding to each of the threads.

39. The processing system of claim 38 wherein the instruction processing unit comprises:

an instruction fetch unit to fetch the instructions from the program memory using a plurality of program counters, each program counter corresponding to each of the threads;

an instruction buffer coupled to the instruction fetch unit to hold the fetched instructions; and

an instruction decoder and dispatcher coupled to the instruction buffer to decode the instructions and dispatch the decoded instructions to one of the memory access unit, the functional unit, and the peripheral unit;

wherein the instructions are executed concurrently in a clock cycle.

40. A processing system comprising:

a plurality of multi-thread processors;

a plurality of peripheral units;

a peripheral bus coupled to the peripheral units to transfer peripheral information between the multi-thread processors and the peripheral units, the peripheral information including a command message sent from one of the multi-thread processors to one of the peripheral units by a thread executing a message instruction.

41. A processing system comprising:

a multi-thread processor having program base registers and data base registers;

at least one peripheral units;

a peripheral bus coupled to the at least one peripheral unit to transfer peripheral information between the multi-thread processor and the at least one peripheral unit, the peripheral information including a command message sent from one of the multi-thread processors to one of the peripheral units by a thread executing a message instruction.

## ABSTRACT OF THE DISCLOSURE

One embodiment of the present invention performs peripheral operations in a multi-thread processor. A peripheral bus is coupled to a peripheral unit to transfer peripheral information including a command message specifying a peripheral operation. A processing slice is coupled to the peripheral bus to execute a plurality of threads. The plurality of threads includes a first thread sending the command message to the peripheral unit.

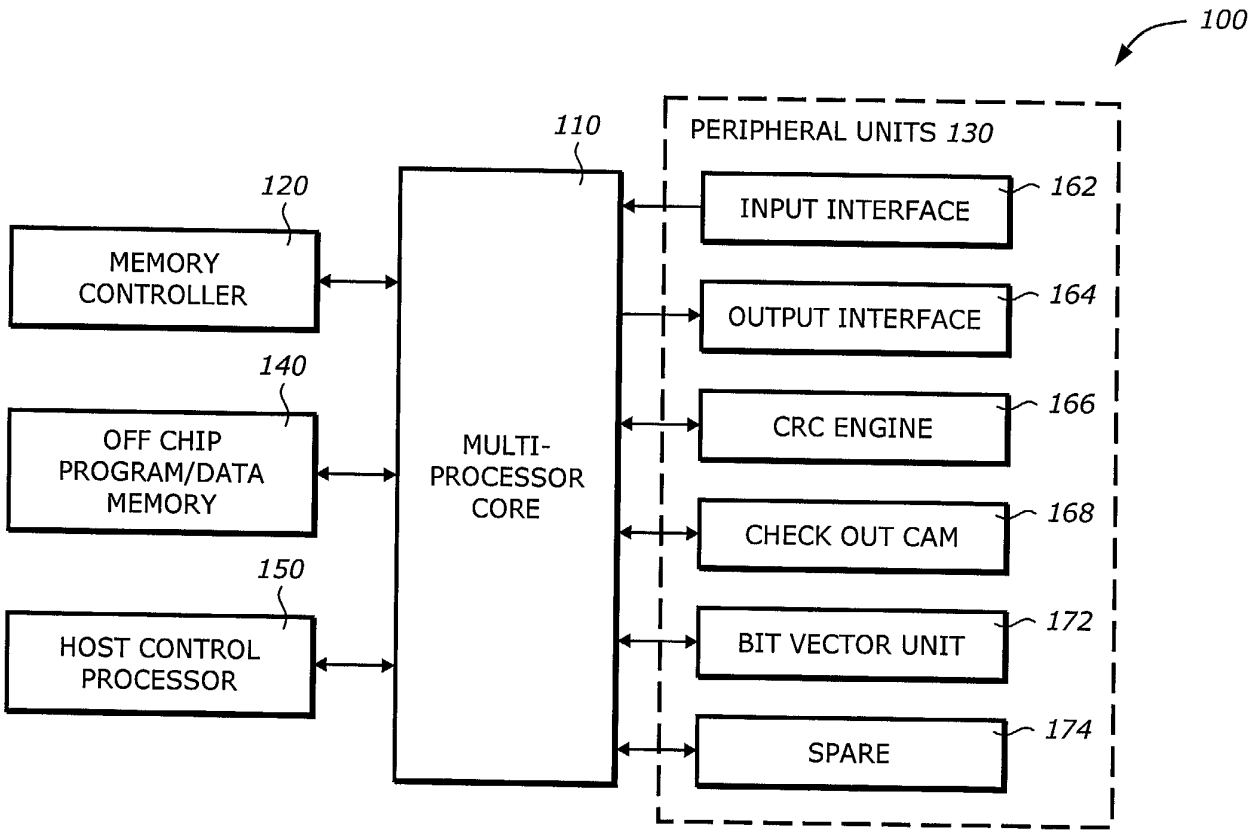

100
110
PERIPHERAL UNITS 130
120
MEMORY CONTROLLER
140
OFF CHIP PROGRAM/DATA MEMORY
150
HOST CONTROL PROCESSOR
MULTI-PROCESSOR CORE
INPUT INTERFACE
162
OUTPUT INTERFACE
164
CRC ENGINE
166
CHECK OUT CAM
168
BIT VECTOR UNIT
172
SPARE
174

FIG. 1

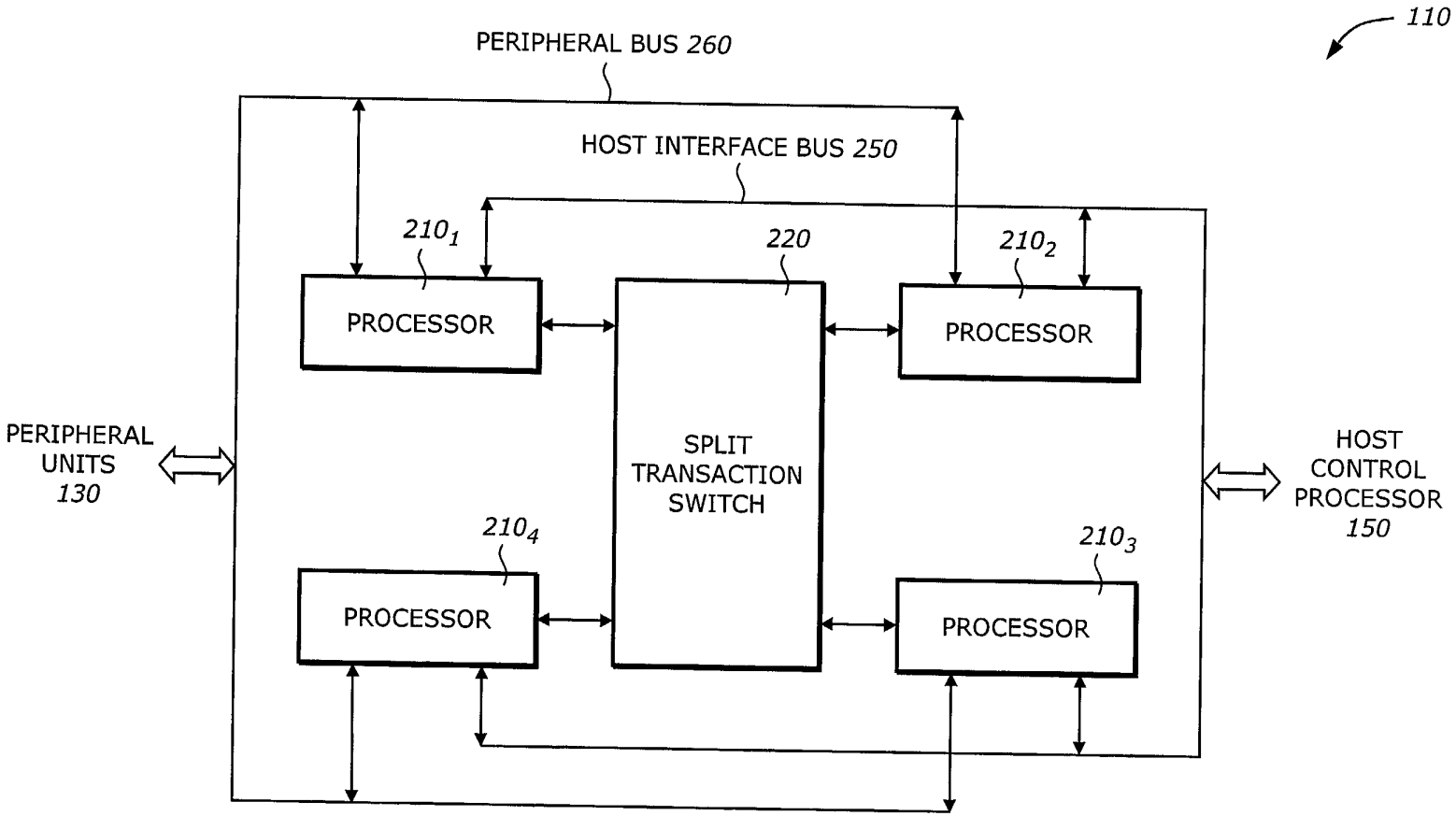



FIG. 2

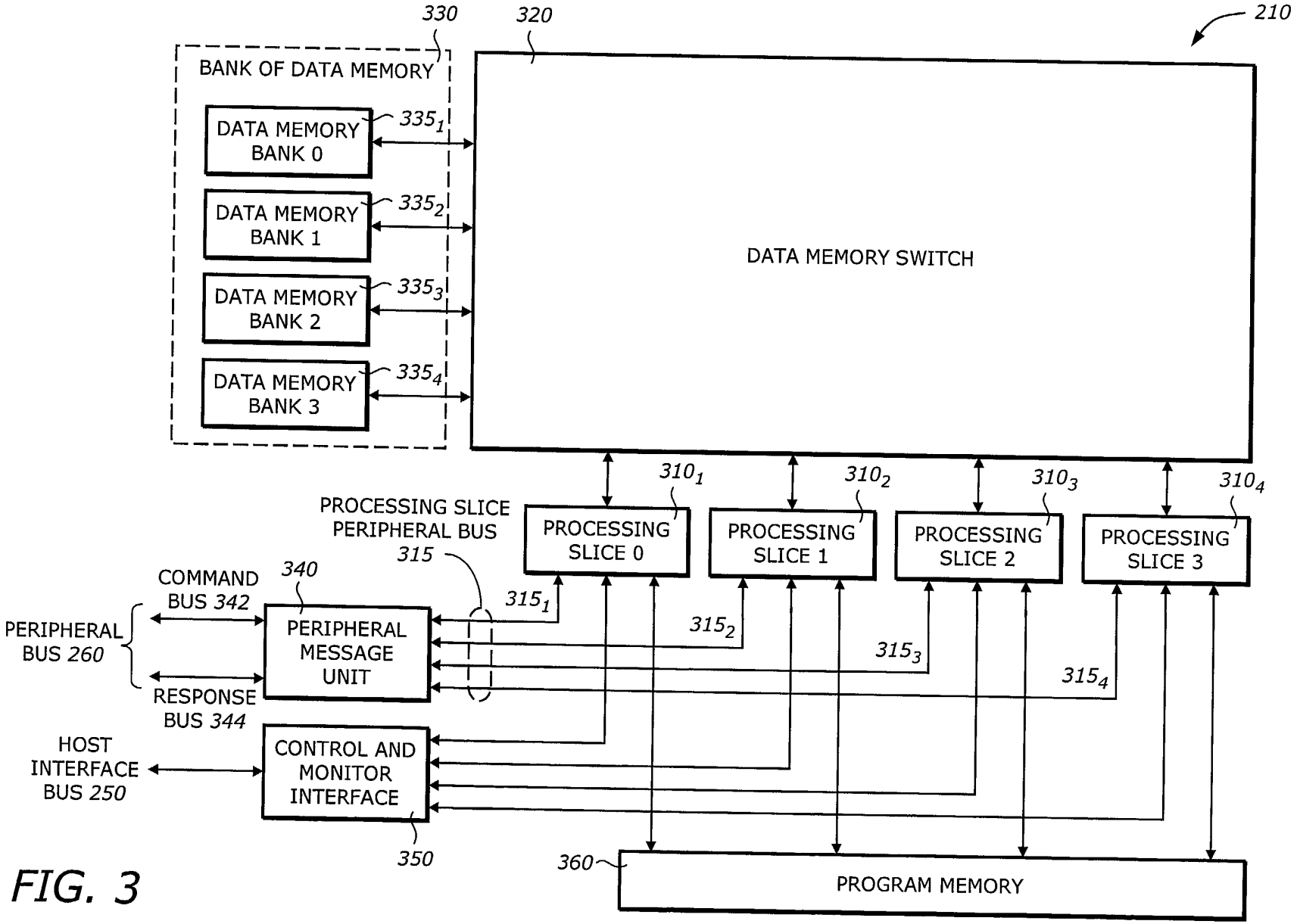

330
320
210
BANK OF DATA MEMORY
DATA MEMORY BANK 0
DATA MEMORY BANK 1
DATA MEMORY BANK 2
DATA MEMORY BANK 3
$335_1$
$335_2$
$335_3$
$335_4$
DATA MEMORY SWITCH
$310_1$
$310_2$
$310_3$
$310_4$
PROCESSING SLICE 0
PROCESSING SLICE 1
PROCESSING SLICE 2
PROCESSING SLICE 3
PROCESSING SLICE PERIPHERAL BUS
315
$315_1$
$315_2$
$315_3$
$315_4$
340
COMMAND BUS 342
PERIPHERAL BUS 260
RESPONSE BUS 344
PERIPHERAL MESSAGE UNIT
HOST INTERFACE BUS 250
CONTROL AND MONITOR INTERFACE
350
360
PROGRAM MEMORY

FIG. 3

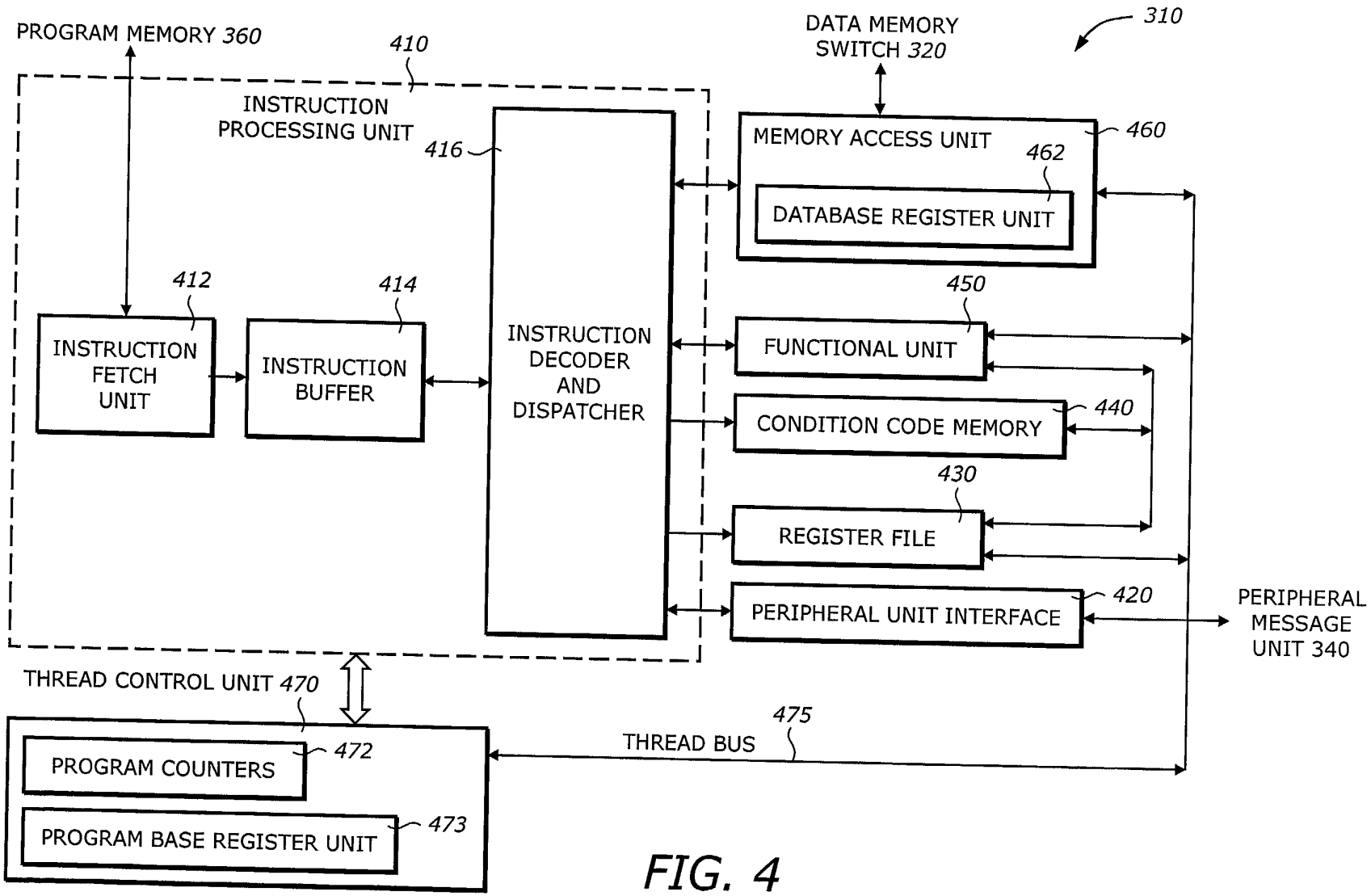

PROGRAM MEMORY 360
410
INSTRUCTION PROCESSING UNIT
416
DATA MEMORY SWITCH 320
310
MEMORY ACCESS UNIT
460
462
DATABASE REGISTER UNIT
412
414
INSTRUCTION FETCH UNIT
INSTRUCTION BUFFER
INSTRUCTION DECODER AND DISPATCHER
450
FUNCTIONAL UNIT
440
CONDITION CODE MEMORY
430
REGISTER FILE
420
PERIPHERAL UNIT INTERFACE
PERIPHERAL MESSAGE UNIT 340
THREAD CONTROL UNIT 470
475
THREAD BUS
472
PROGRAM COUNTERS
473
PROGRAM BASE REGISTER UNIT

FIG. 4

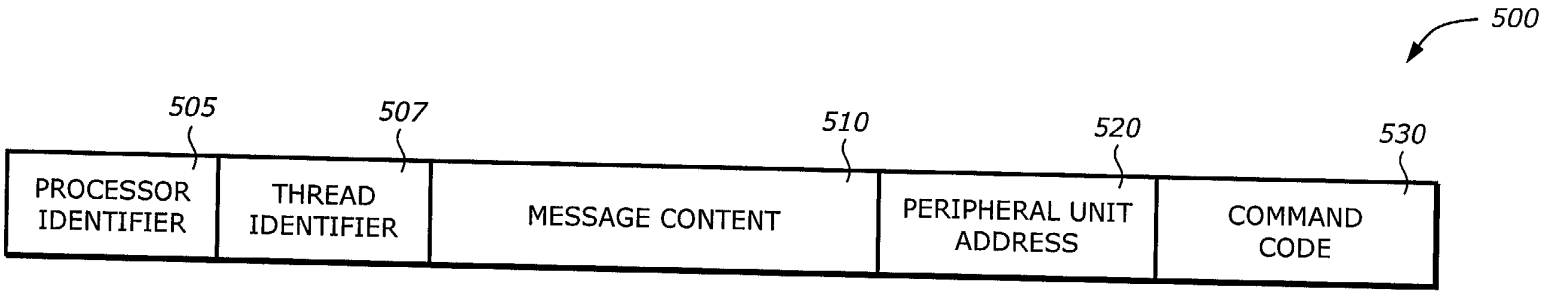



FIG. 5A

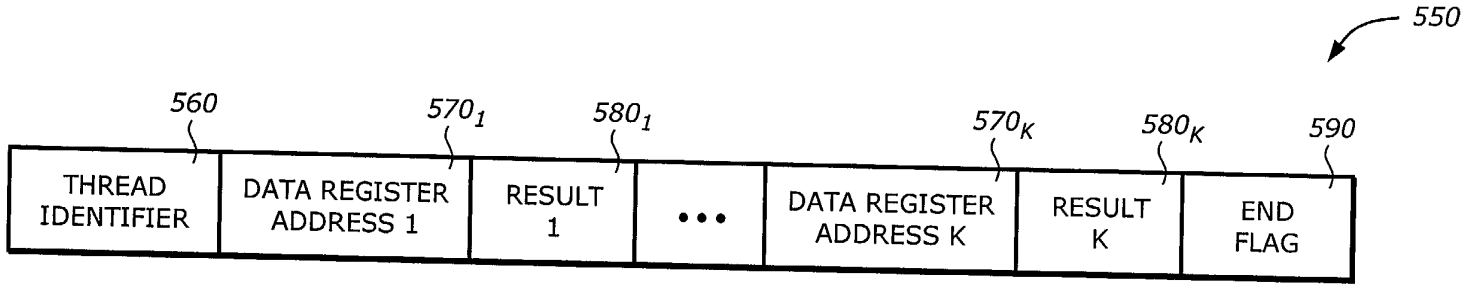



FIG. 5B

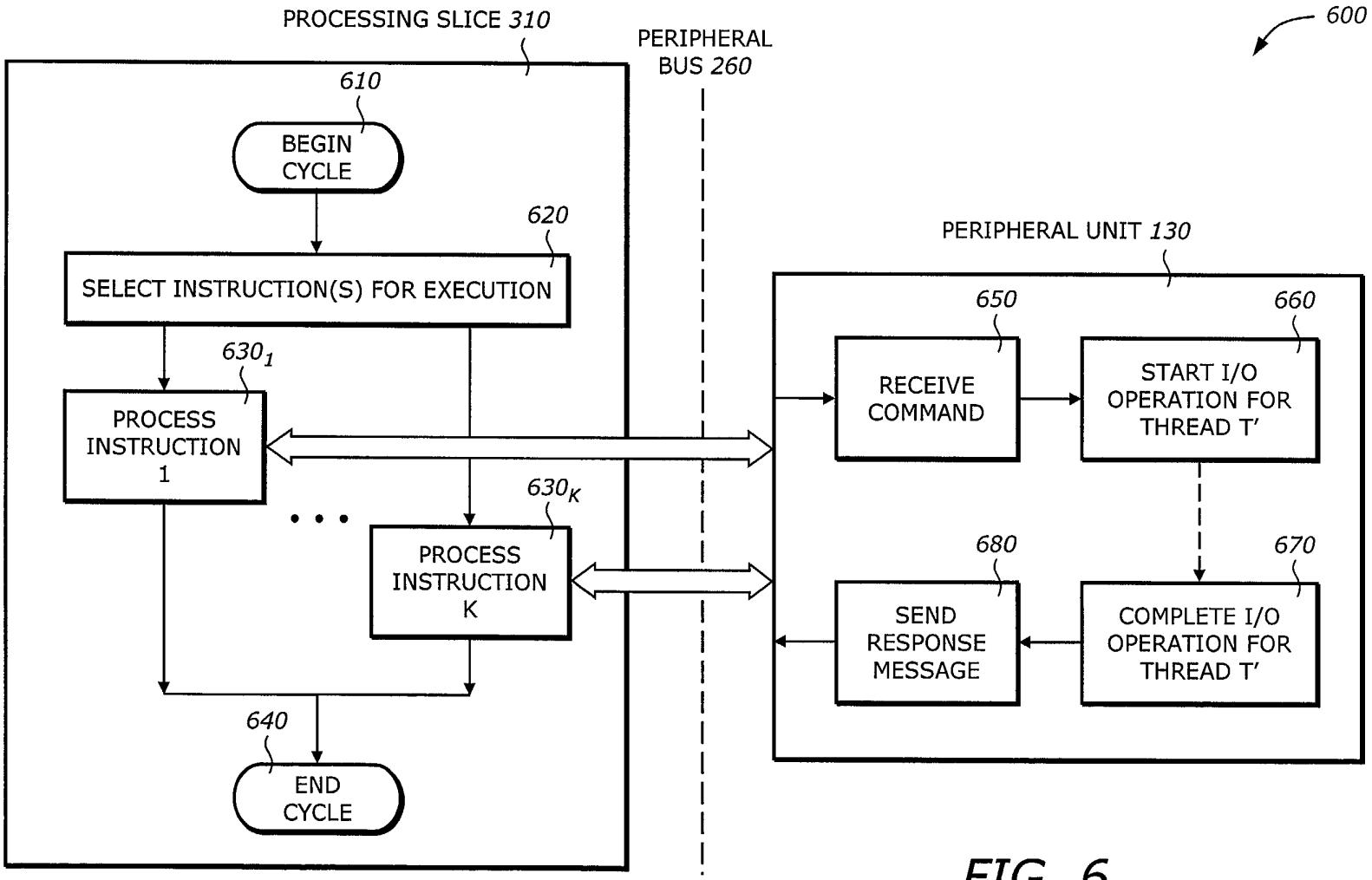



FIG. 6

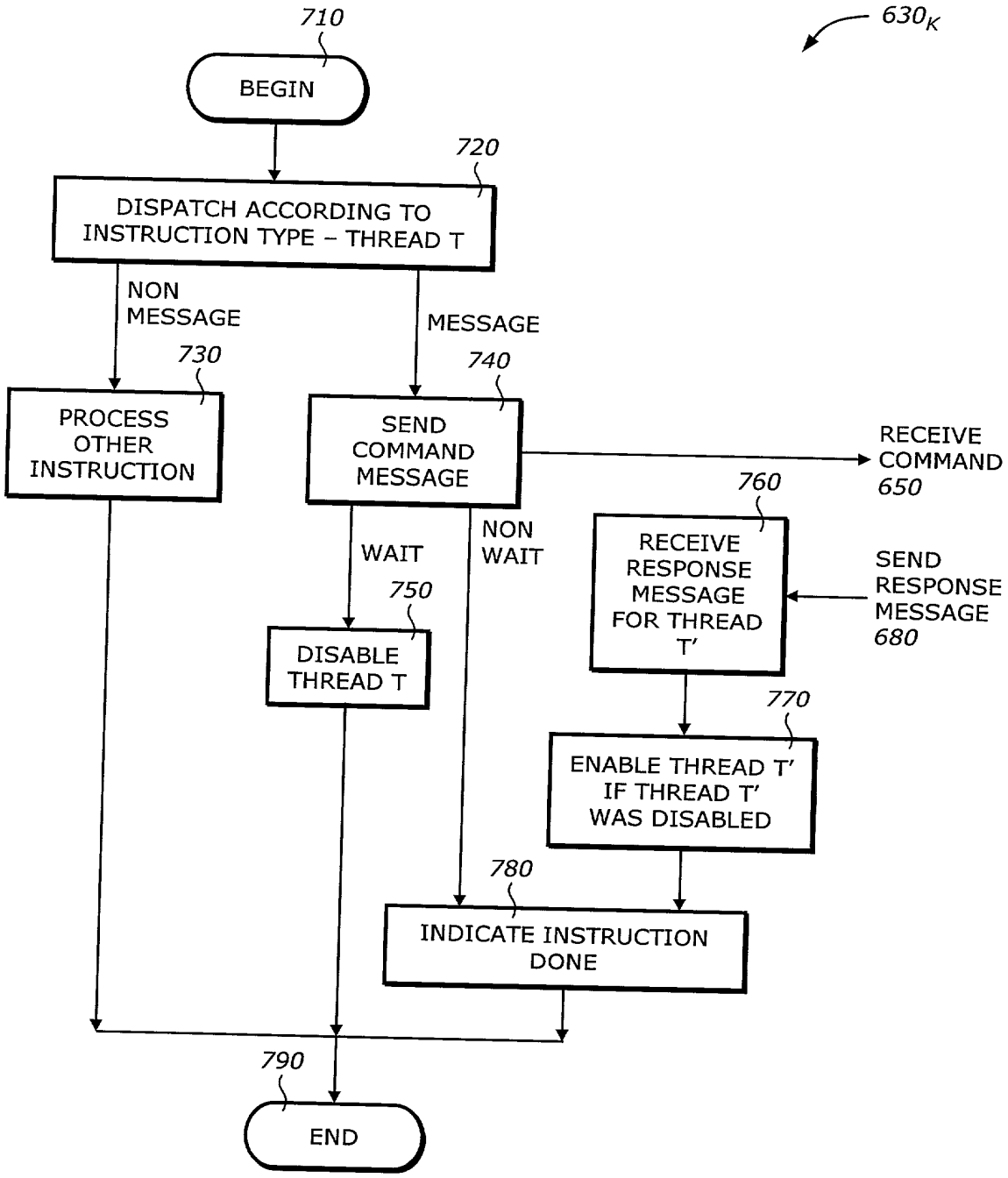

630K
710
BEGIN
720
DISPATCH ACCORDING TO INSTRUCTION TYPE – THREAD T
NON MESSAGE
MESSAGE
730
PROCESS OTHER INSTRUCTION
740
SEND COMMAND MESSAGE
RECEIVE COMMAND 650
760
RECEIVE RESPONSE MESSAGE FOR THREAD T'
SEND RESPONSE MESSAGE 680
WAIT
NON WAIT
750
DISABLE THREAD T
770
ENABLE THREAD T' IF THREAD T' WAS DISABLED
780
INDICATE INSTRUCTION DONE
790
END

FIG. 7

Attorney's Docket No.: 004800.P004

# DECLARATION AND POWER OF ATTORNEY FOR PATENT APPLICATION

As a below named inventor, I hereby declare that:

My residence, post office address and citizenship are as stated below, next to my name.

I believe I am the original, first, and sole inventor (if only one name is listed below) or any original, first, and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

## Multi-Thread Peripheral Processing Using Dedicated Peripheral Bus

the specification of which [X] is attached hereto.
[ ] was filed on _____ as
United States Application Number _____
or PCT International Application Number _____
and was amended on _____
(if applicable)

I hereby state that I have reviewed and understand the contents of the above-identified specification, including the claim(s), as amended by any amendment referred to above. I do not know and do not believe that the claimed invention was ever known or used in the United States of America before my invention thereof, or patented or described in any printed publication in any country before my invention thereof or more than one year prior to this application, that the same was not in public use or on sale in the United States of America more than one year prior to this application, and that the invention has not been patented or made the subject of an inventor's certificate issued before the date of this application in any country foreign to the United States of America on an application filed by me or my legal representatives or assigns more than twelve months (for a utility patent application) or six months (for a design patent application) prior to this application.

I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

I hereby claim foreign priority benefits under Title 35, United States Code, Section 119(a)-(d), of any foreign application(s) for patent or inventor's certificate listed below and have also identified below any foreign application for patent or inventor's certificate having a filing date before that of the application on which priority is claimed:

Prior Foreign Application(s):

| APPLICATION NUMBER | COUNTRY (OR INDICATE IF PCT) | DATE OF FILING (day, **month**, year) | PRIORITY CLAIMED UNDER 37 USC 119 |
|---|---|---|---|
| | | | ☐ No ☐ Yes |
| | | | ☐ No ☐ Yes |
| | | | ☐ No ☐ Yes |

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below:

| APPLICATION NUMBER | FILING DATE |
|---|---|
| 60/166,686 | November 19, 1999 |
| | |

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s) listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States application in the manner provided by the first paragraph of Title 35, United States Code, Section 112, I acknowledge the duty to disclose all information known to me to be material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of this application:

| APPLICATION NUMBER | FILING DATE | STATUS (ISSUED, PENDING, ABANDONED) |
|---|---|---|
| | | |
| | | |
| | | |

I hereby appoint the persons listed on Appendix A hereto (which is incorporated by reference and a part of this document) as my respective patent attorneys and patent agents, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.

Send correspondence to:
Thinh V. Nguyen, Reg. No. 42,034, BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP
(Name of Attorney or Agent)
12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025 and direct telephone calls to:
Thinh V. Nguyen, (714) 557-3800.
(Name of Attorney or Agent)

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

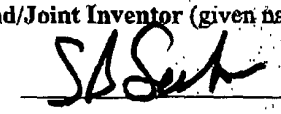
**Full Name of Sole/First Inventor** (given name, family name) **Jack B. Dennis**

Inventor's Signature  Date 17 November 2000

Residence Belmont, Massachusetts USA (*City, State*) Citizenship USA (*Country*)

Mailing Address 55 Wellesley Road
Belmont, Massachusetts 02478 USA

**Full Name of Second/Joint Inventor** (given name, family name) **Sam B. Sandbote**

Inventor's Signature  Date 11/17/2000

Residence Reston, Virginia USA (*City, State*) Citizenship USA (*Country*)

Mailing Address 1988 Crescent Park Drive
Reston, Virginia 20190 USA

# Appendix A

I hereby appoint BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP, a firm including: William E. Alford, Reg. No. 37,764; Farzad E. Amini, Reg. No. 42,261; William Thomas Babbitt, Reg. No. 39,591; Carol F. Barry, Reg. No. 41,600; Jordan Michael Becker, Reg. No. 39,602; Lisa N. Benado, Reg. No. 39,995; Bradley J. Bereznak, Reg. No. 33,474; Michael A. Bernadicou, Reg. No. 35,934; Roger W. Blakely, Jr., Reg. No. 25,831; R. Alan Burnett, Reg. No. 46,149; Gregory D. Caldwell, Reg. No. 39,926; Andrew C. Chen, Reg. No. 43,544; Thomas M. Coester, Reg. No. 39,637; Donna Jo Coningsby, Reg. No. 41,684; Dennis M. deGuzman, Reg. No. 41,702; Stephen M. De Klerk, Reg. No. P46,503; Michael Anthony DeSanctis, Reg. No. 39,957; Daniel M. De Vos, Reg. No. 37,813; Sanjeet Dutta, Reg. No. P46,145; Matthew C. Fagan, Reg. No. 37,542; Tarek N. Fahmi, Reg. No. 41,402; George Fountain, Reg. No. 36,374; Paramita Ghosh, Reg. No. 42,806; James Y. Go, Reg. No. 40,621; James A. Henry, Reg. No. 41,064; Willmore F. Holbrow III, Reg. No. P41,845; Sheryl Sue Holloway, Reg. No. 37,850; George W Hoover II, Reg. No. 32,992; Eric S. Hyman, Reg. No. 30,139; William W. Kidd, Reg. No. 31,772; Sang Hui Kim, Reg. No. 40,450; Walter T. Kim, Reg. No. 42,731; Eric T. King, Reg. No. 44,188; Erica W. Kuo, Reg. No. 42,775; George B. Leavell, Reg. No. 45,436; Gordon R. Lindeen III, Reg. No. 33,192; Jan Carol Little, Reg. No. 41,181; Kurt P. Leyendecker, Reg. No. 42,799; Joseph Lutz, Reg. No. 43,765; Michael J. Mallie, Reg. No. 36,591; Andre L. Marais, under 37 C.F.R. § 10.9(b); Paul A. Mendonsa, Reg. No. 42,879; Clive D. Menezes, Reg. No. 45,493; Chun M. Ng, Reg. No. 36,878; Thien T. Nguyen, Reg. No. 43,835; Thinh V. Nguyen, Reg. No. 42,034; Dennis A. Nicholls, Reg. No. 42,036; Daniel E. Ovanezian, Reg. No. 41,236; Kenneth B. Paley, Reg. No. 38,989; Marina Portnova, Reg. No. P45,750; William F. Ryann, Reg. 44,313; James H. Salter, Reg. No. 35,668; William W. Schaal, Reg. No. 39,018; James C. Scheller, Reg. No. 31,195; Jeffrey S. Schubert, Reg. No. 43,098; Jeffrey Sam Smith, Reg. No. 39,377; Maria McCormack Sobrino, Reg. No. 31,639; Stanley W. Sokoloff, Reg. No. 25,128; Judith A. Szepesi, Reg. No. 39,393; Vincent P. Tassinari, Reg. No. 42,179; Edwin H. Taylor, Reg. No. 25,129; John F. Travis, Reg. No. 43,203; Joseph A. Twarowski, Reg. No. 42,191; Thomas A. Van Zandt, Reg. No. 43,219; Lester J. Vincent, Reg. No. 31,460; Glenn E. Von Tersch, Reg. No. 41,364; John Patrick Ward, Reg. No. 40,216; Mark L. Watson, Reg. No. P46,322; Thomas C. Webster, Reg. No. P46,154; and Norman Zafman, Reg. No. 26,250; my patent attorneys, and Firasat Ali, Reg. No. 45,715; and Justin M. Dillon, Reg. No. 42,486; Raul Martinez, Reg. No. 46,904; my patent agents, with offices located at 12400 Wilshire Boulevard, 7th Floor, Los Angeles, California 90025, telephone (714) 557-3800, with full power of substitution and revocation, to prosecute this application and to transact all business in the Patent and Trademark Office connected herewith.